



**Università
degli Studi
di Palermo**



Introduzione alla Programmazione

CALCOLATORI ELETTRONICI – FONDAMENTI DI PROGRAMMAZIONE
a.a. 2024/2025

Prof. Roberto Pirrone

Sommario

- Ciclo di vita del software
- Rappresentazione degli algoritmi
- Creazione di un programma



Ciclo di vita del software

1. Prima dell'implementazione
 1. Studio di fattibilità
 2. Descrizione del problema
 3. Progetto del programma
 4. Scelta o sviluppo dell'algoritmo e analisi
2. Implementazione
 1. Codifica
 2. Debugging
3. Dopo l'implementazione
 1. Testing, verifica e valutazione delle prestazioni
 2. Documentazione
 3. Manutenzione

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

Dialogo con l'utente che espone le sue necessità, interpretazione di massima delle specifiche e primo documento di analisi.

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

Definizione delle specifiche per mezzo di tecniche formali. Stesura di un documento di analisi dei requisiti da sottoporre all'utente.

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

Progettazione in grande:

Si analizza la suddivisione del software per componenti funzionali e lo scambio dati tra questi ultimi.

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

Progettazione in piccolo:
Per ogni componente si fa un'analisi di dettaglio e si definisce l'algoritmo.

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

2. Implementazione

1. Codifica
2. Debugging

Scelta del linguaggio di programmazione e sviluppo delle singole parti con azione di correzione degli errori (prototipi).

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Di norma un team diverso da quello di progetto. A questo punto va fatta anche l'integrazione tra le parti.

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Può essere scritta anche durante lo sviluppo, man mano che si completa il software.

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione

Effettuata dopo la distribuzione del prodotto, con l'aiuto degli utenti: nuove release e/o versioni.

Ciclo di vita del software

1. Prima dell'implementazione

1. Studio di fattibilità
2. Descrizione del problema
3. Progetto del programma
4. Scelta o sviluppo dell'algoritmo e analisi

2. Implementazione

1. Codifica
2. Debugging

3. Dopo l'implementazione

1. Testing, verifica e valutazione delle prestazioni
2. Documentazione
3. Manutenzione



E se le cose non vanno bene?

Rappresentazione degli Algoritmi

- Le fasi di progettazione che precedono la codifica di un algoritmo, che diventa programma, richiedono un livello di formalizzazione crescente
 - Descrizione in linguaggio naturale
 - Pseudocodice e/o rappresentazione diagrammatica
 - Codifica in un linguaggio di programmazione

Rappresentazione degli Algoritmi

- Problema: sommiamo due numeri interi in colonna
- **Linguaggio naturale**
 - La somma di due numeri di m cifre si calcola addizionando, posizione per posizione, le cifre omologhe dei due addendi con l'eventuale riporto proveniente dalla colonna precedente; se la somma in una posizione ha valore maggiore o uguale a dieci allora si scriverà la sola cifra delle unità nel risultato mentre il valore uno di riporto verrà addizionato con le cifre della colonna successiva.

Rappresentazione degli Algoritmi

- Problema: sommiamo due numeri interi in colonna
- **Pseudocodice**
 - Input: m numero di colonne
 $a_0a_1...a_{m-1}$ e $b_0b_1...b_{m-1}$ le due sequenze di cifre da sommare
 - Output: $c_0c_1...c_{m-1}c_m$ la sequenza di cifre del risultato

Rappresentazione degli Algoritmi

- Problema: sommiamo due numeri interi in colonna
- **Pseudocodice**
 1. Imposta il valore di *riporto* a 0
 2. Ripeti per ogni cifra degli addendi
 3. Somma a_i e b_i con *riporto* per ottenere c_i
 4. Se c_i è maggiore o uguale a 10
 5. Imposta il valore di c_i a $c_i - 10$
 6. Imposta il valore di *riporto* a 1
 7. Altrimenti poni *riporto* uguale a 0

Rappresentazione degli Algoritmi

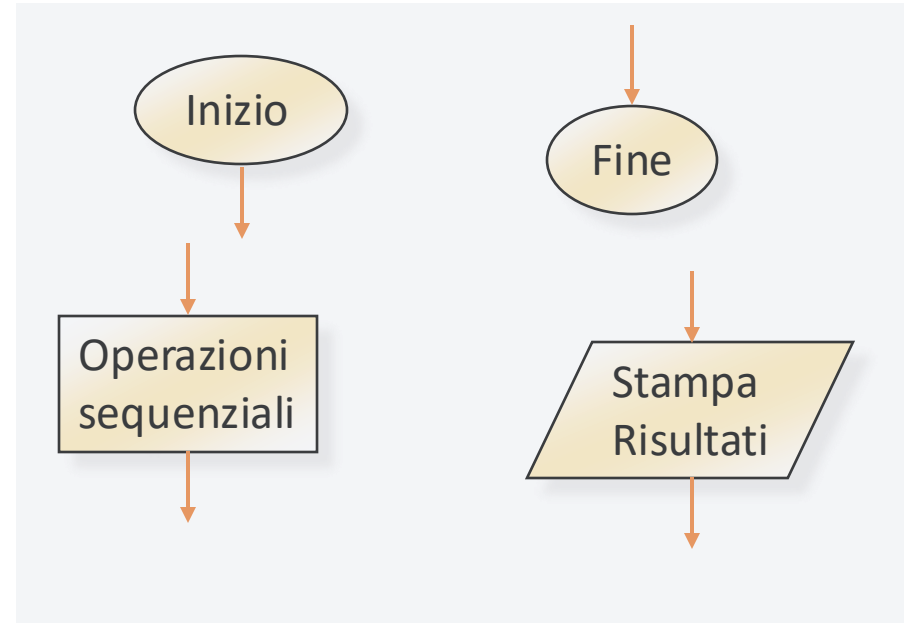
- Problema: sommiamo due numeri interi in colonna

- **Pseudocodice**

2. *Ripeti* per ogni cifra degli addendi
3. Somma a_i e b_i con *riporto* per ottenere c_i
4. *Se* c_i è maggiore o uguale a 10
5. Imposta il valore di c_i a $c_i - 10$
6. Imposta il valore di *riporto* a 1
7. *Altrimenti* poni *riporto* uguale a 0

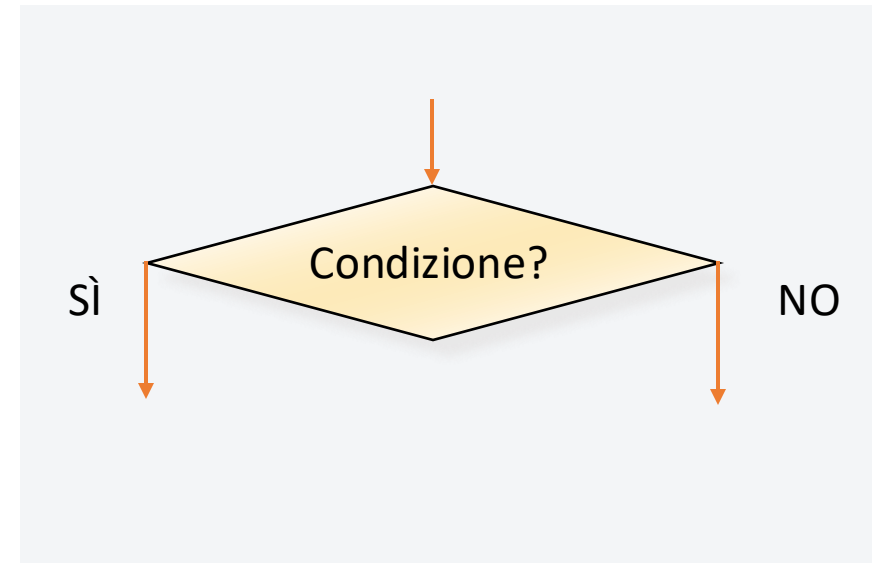
Rappresentazione degli Algoritmi

- Notazione diagrammatica:
Diagrammi di Flusso
- Operazioni sequenziali
 - Realizza un solo compito ben definito
 - Il controllo passa all'operazione successiva quando il compito è finito
 - Frase dichiarativa



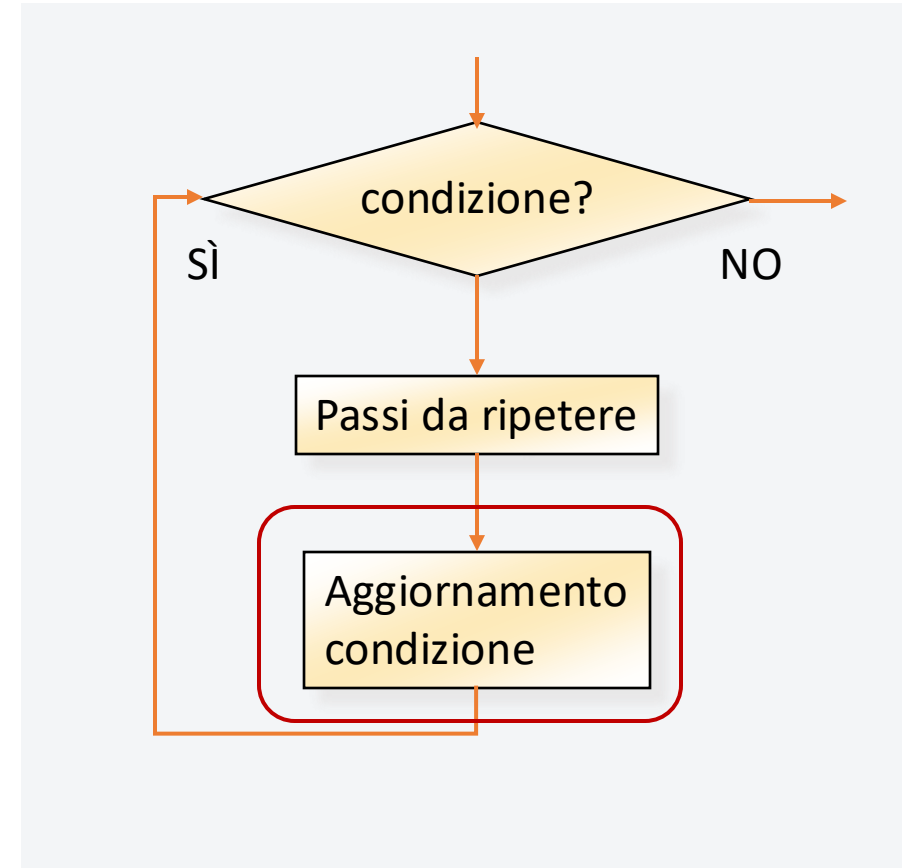
Rappresentazione degli Algoritmi

- Notazione diagrammatica:
Diagrammi di Flusso
- Operazioni condizionali
 - Selezionano l'operazione successiva sulla base di una domanda



Rappresentazione degli Algoritmi

- Notazione diagrammatica:
Diagrammi di Flusso
- Operazioni iterative
 - Eseguono un ciclo di istruzioni finché la condizione di controllo rimane verificata
 - La condizione deve essere «cambiata» dall'algoritmo perché a un certo punto deve diventare falsa



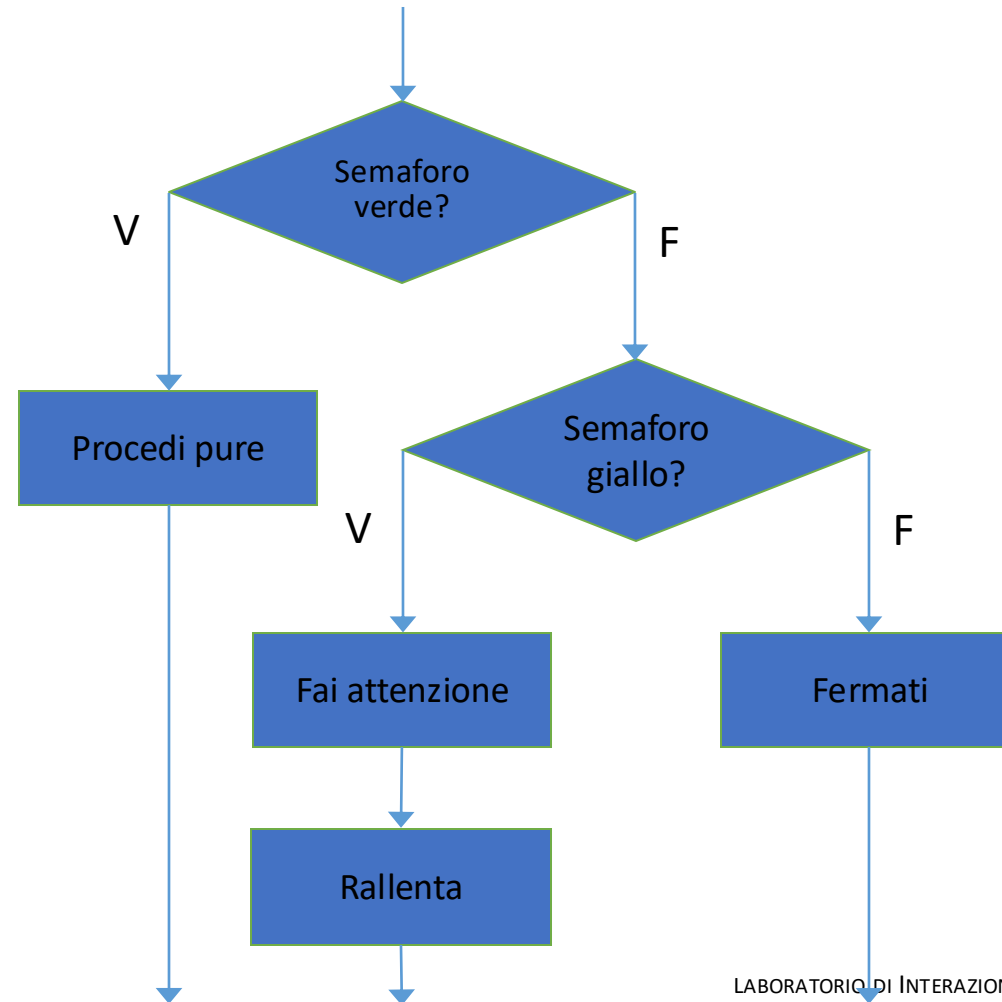
Rappresentazione degli Algoritmi

- Posso inserire più istruzioni di controllo una dentro l'altra
- L'algoritmo del semaforo
 - Se il semaforo è verde procedi pure, se è giallo fai attenzione e rallenta, se è rosso fermati.
 - L'algoritmo funziona sulla base del test "di che colore è il semaforo?" che **non ha** una risposta vero/falso, ma una risposta triplice.

Rappresentazione degli Algoritmi

Scelta multipla tra più condizioni:

1. Se il semaforo è verde
 1. Procedi pure
2. Altrimenti se è giallo
 1. Fai attenzione
 2. Rallenta
3. Altrimenti
 1. Fermati

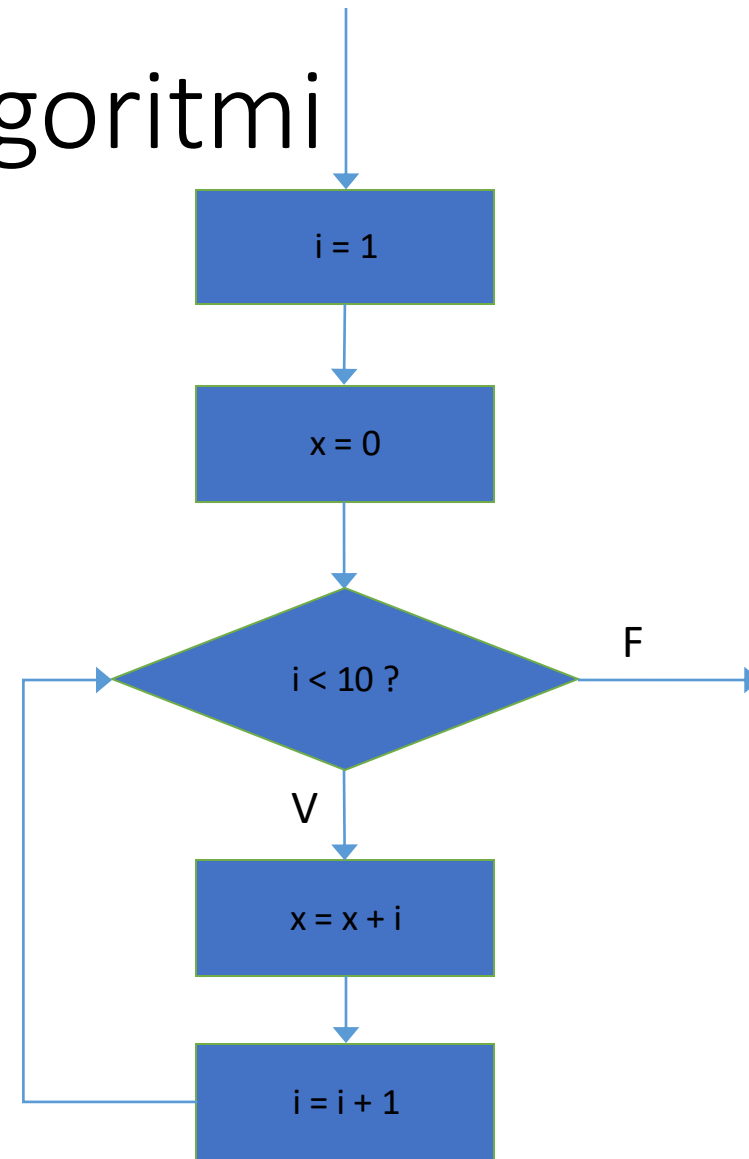


Rappresentazione degli Algoritmi

Istruzioni Iterative

1. Poni $i = 1$
2. Poni $x = 0$
3. Ripeti mentre $i < 10$
 1. Poni $x = x + i$
 2. Poni $i = i + 1$

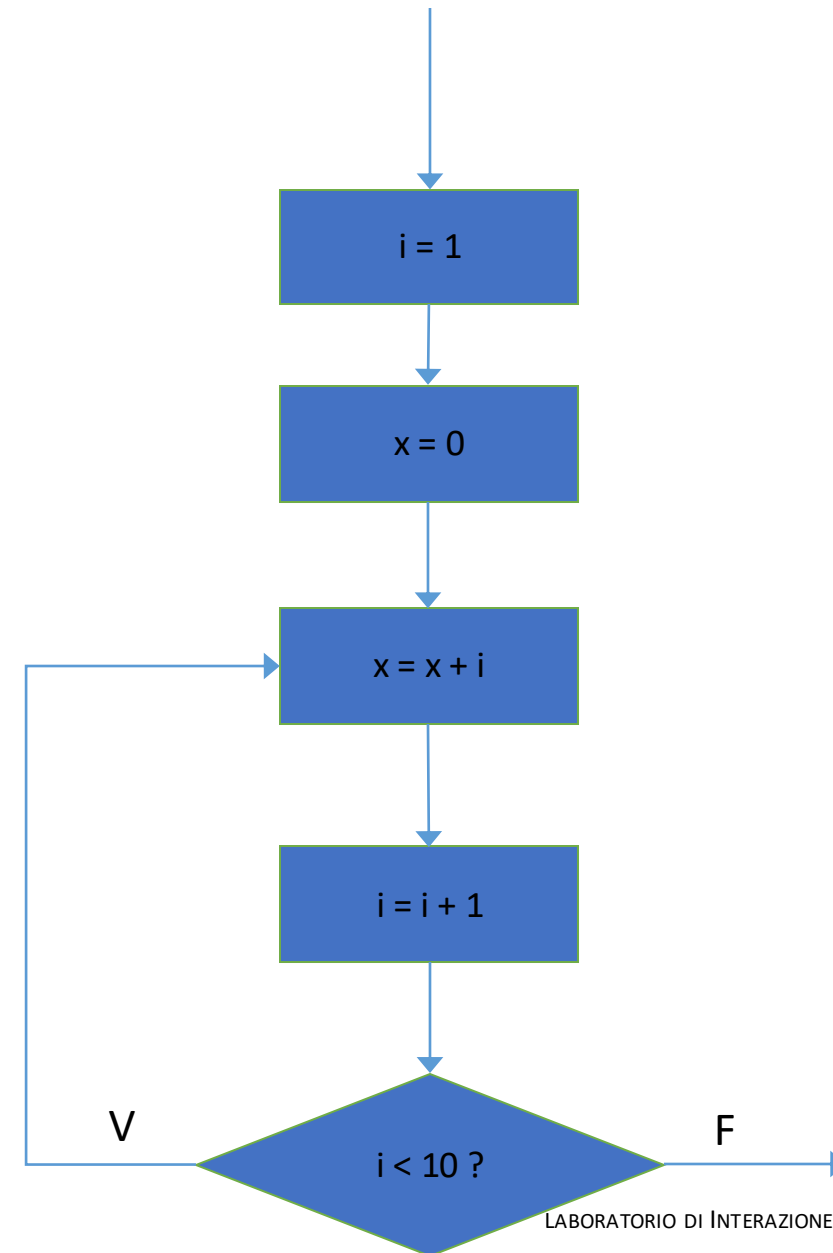
Che succede se in 1. poniamo $i = 11$?



Istruzioni iterative

1. Poni $i = 1$
2. Poni $x = 0$
3. Ripeti
 1. Poni $x = x + i$
 2. Poni $i = i + 1$
4. mentre $i < 10$

Che succede se in 1.
poniamo $i = 11$?

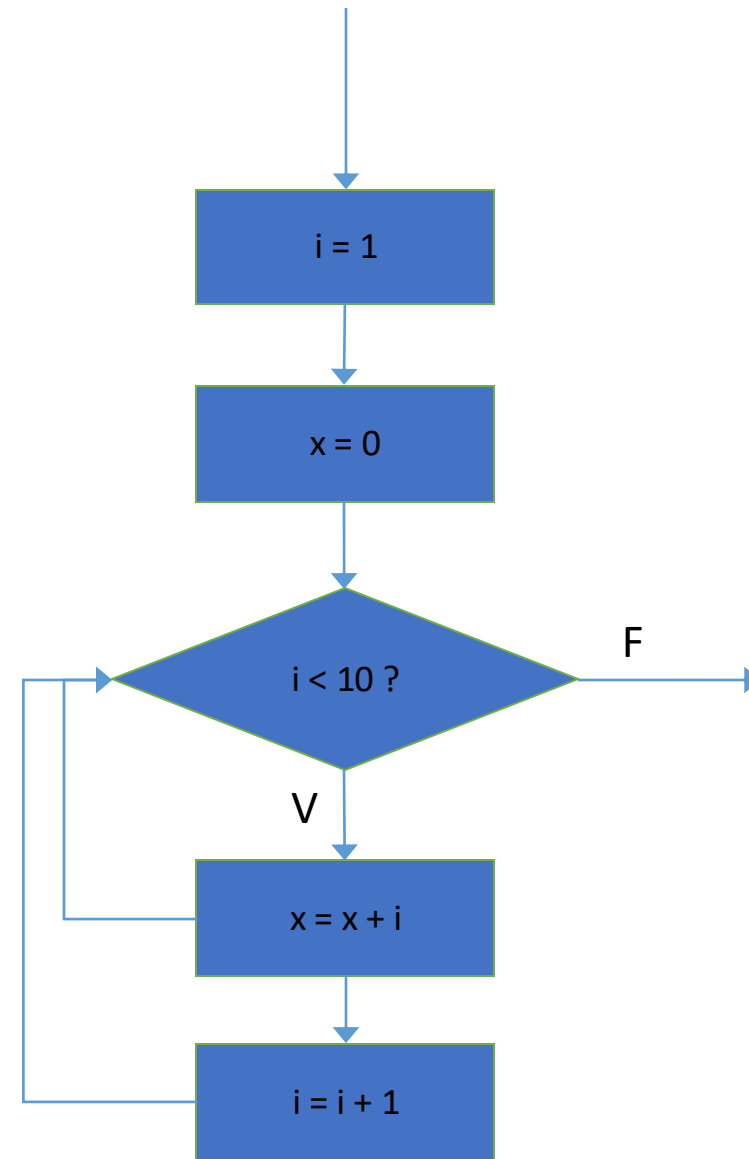


Istruzioni iterative

1. Poni $i = 1$
2. Poni $x = 0$
3. Ripeti mentre $i < 10$
 1. Poni $x = x + i$
 2. Poni $i = i + 1$

Numero finito di passi, ma
esecuzione infinita:

NON E' UN ALGORITMO

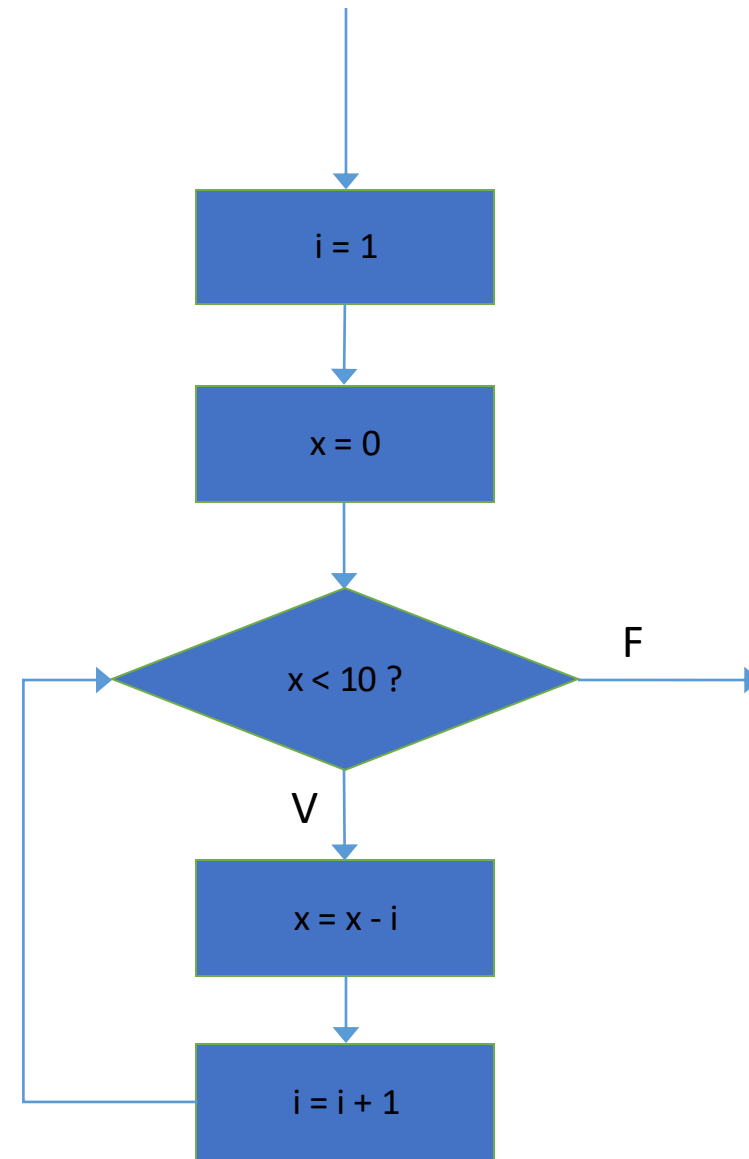


Istruzioni iterative

1. Poni $i = 1$
2. Poni $x = 0$
3. Ripeti mentre $x < 10$
 1. Poni $x = x - i$
 2. Poni $i = i + 1$

Ciclo infinito!!

NON E' UN ALGORITMO



Rappresentazione degli Algoritmi

- Problema: sommiamo due numeri interi in colonna

- **Pseudocodice**

1. Imposta il valore di *riporto* a 0
2. Ripeti per ogni cifra degli addendi
 3. Somma a_i e b_i con *riporto* per ottenere c_i
 4. Se c_i è maggiore o uguale a 10
 5. Imposta il valore di c_i a $c_i - 10$
 6. Imposta il valore di *riporto* a 1
 7. Altrimenti poni *riporto* uguale a 0
8. Imposta c_m a *riporto*
9. Stampa $c_0 c_1 \dots c_m$
10. Fine

Rappresentazione degli Algoritmi

La somma di due numeri:

1

47 +

25 =

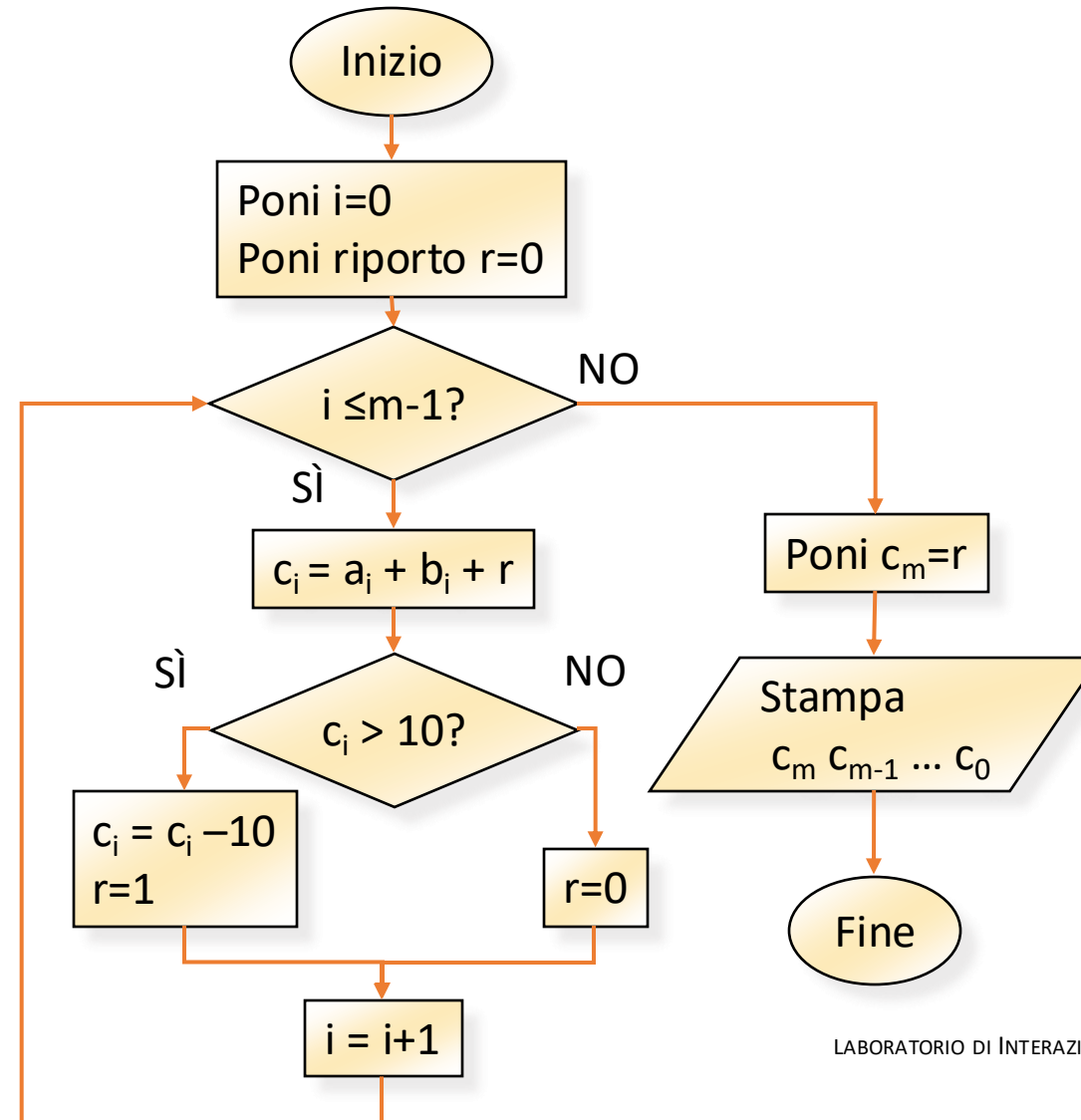
72

Due numeri con $m > 1$ cifre:

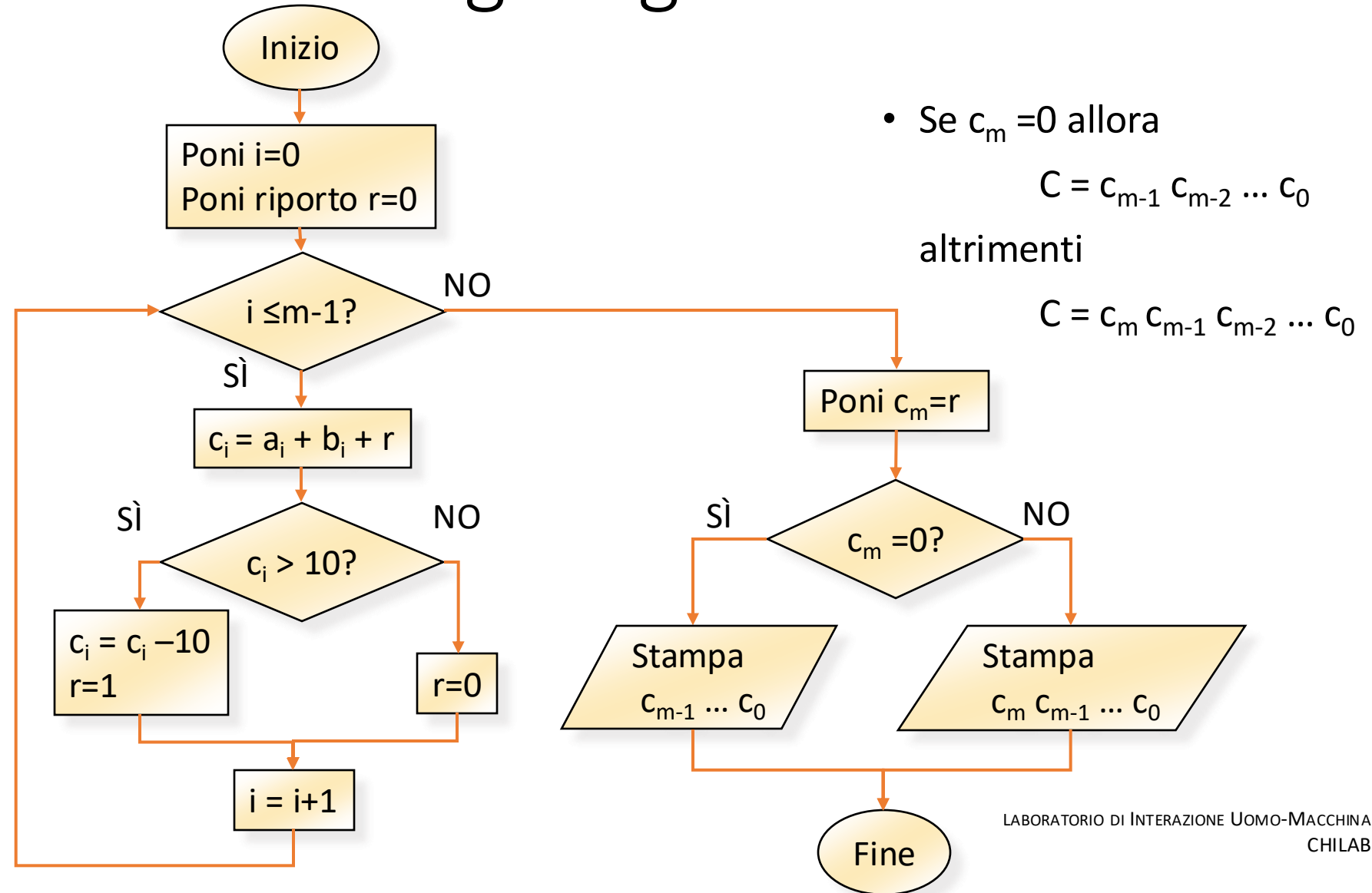
A = $a_{m-1} a_{m-2} \dots a_0 +$

B = $b_{m-1} b_{m-2} \dots b_0 =$

C = $c_m c_{m-1} c_{m-2} \dots c_0$



Rappresentazione degli Algoritmi



Rappresentazione degli Algoritmi

```
#include <stdio.h>
```

```
int main() {  
    // Cifre del primo numero  
    int a1, a2, a3;  
    // Cifre del secondo numero  
    int b1, b2, b3;  
  
    // Richiesta delle cifre del primo numero  
    printf("Inserisci le cifre del primo numero (3 cifre separate da spazi): ");  
    scanf("%d %d %d", &a1, &a2, &a3);  
  
    // Richiesta delle cifre del secondo numero  
    printf("Inserisci le cifre del secondo numero (3 cifre separate da spazi): ");  
    scanf("%d %d %d", &b1, &b2, &b3);  
  
    // Variabili per la somma e il riporto  
    int s1, s2, s3, riporto = 0;  
  
    // Somma delle unità  
    s3 = a3 + b3 + riporto;  
    riporto = s3 / 10;  
    s3 = s3 % 10;  
  
    // Somma delle decine  
    s2 = a2 + b2 + riporto;  
    riporto = s2 / 10;  
    s2 = s2 % 10;  
  
    // Somma delle centinaia  
    s1 = a1 + b1 + riporto;  
    riporto = s1 / 10;  
    s1 = s1 % 10;  
  
    // Stampa del risultato  
    if (riporto > 0) {  
        printf("La somma è: %d%d%d%d\n", riporto, s1, s2, s3);  
    } else {  
        printf("La somma è: %d%d%d\n", s1, s2, s3);  
    }  
  
    return 0;  
}
```